

### **Remarks**

Applicants respectfully request reconsideration of the present application in view of the foregoing amendments and the following remarks. Claims 1, 3-7, 30-34, and 38-43 are pending in the application. Claims 1, 3-8, 10, 11, 17, 19, and 30-37 are rejected. No claims have been allowed. Claims 1, 7, 38, and 43 are independent. Various amendments are made for editorial purposes and not in response to the art. Canceled claims are canceled without disclaimer or prejudice to renewal.

### ***Claim Amendments***

The amendments herein find support in the original application. Exemplary support for “the representations of state form a chain of state frames” is found at page 13, lines 15 *et seq.*: “Thus, the new frame 504 does not require a copy of the universe of variables or data since it is available by walking back the frame chain.”

Exemplary support for “at least one of the state frames in the chain of state frames comprises a cached value copied from a prior frame in the chain of state frames” is found at page 4, lines 19 *et seq.*:

To speed-up further accesses on the same variable, a cache of the values of variables can be established. In one example, this cache can be realized by copying an update to the present frame from a prior frame when an access occurs. Thus, the next time the same variable is accessed, a walk backwards will only be necessary until the frame is reached where either the variable has been originally updated or has been last accessed.

Exemplary support for amendments to claim 7 are found in FIG. 9 and the accompanying description at pages 16-18. For example, page 17, lines 5-13 describe:

While a program or model executes 902, a state save request is made 904. If the state save request is the first state save for this program execution (e.g., an origin state save) 912, the entire state of the executing program is saved 914, and a new present frame is created 916 to hold future updates to this saved frame. If an origin frame already exists, then a present frame is saved as the saved frame 918, and a new present frame is created 916 to save updates occurring by the executing program. Whether or not an origin save frame is created, the new present frame has a back

pointer to the saved frame 920. As the program continues 922, the state changes of the program are stored as updates in the present frame 916.

### Other Claims

New claim 38 mimics claim 7.

Claim 39 finds exemplary support at page 20, line 12 of the Application.

Claim 40 finds exemplary support at page 22, line 1 of the Application.

Claim 41 finds exemplary support at page 22, line 10 of the Application.

Claim 42 finds exemplary support at page 14, lines 11 *et seq.* of the Application.

The “specialized computer” of new claim 43 finds support at page 27, line 20.

Accordingly, no new matter is added thereby.

Separately, an amendment is made in the specification to correct a typographical error, and no new matter is added thereby.

### *Cited Art*

The Action cites Baum et al., U.S. Patent No. 5,220,669 (hereinafter “Baum”).

### *Claim Rejections under 35 U.S.C. § 102*

The Action rejects claims 1, 3-8, 10, 11, 17, 19, and 30-37 under 35 USC 102(b) as being anticipated by Baum. Applicants respectfully submit the claims are allowable over the cited art. For a 102(b) rejection to be proper, the cited art must show each and every element as set forth in a claim. (*See* MPEP § 2131.01.) However, the cited art does not describe each and every element. Accordingly, Applicants request that all rejections be withdrawn.

### Claim 1

**Baum’s description of features related to a linkage stack does not anticipate the recited “chain of state frames,” and particularly “checking, in backward order . . . for the value of the variable in the chain of state frames.”** The Action points to passages in Baum as anticipating the recited arrangement. For example, Baum describes

at column 15, lines 61 *et seq.*:

FIG. 9 shows a linkage stack 35 which may be formed by the control program for each dispatchable unit. The linkage stack is used to save the execution state and the contents of the general-registers and access-registers during a stacking operation. The linkage stack is also used to restore a portion of the execution state and the general-register and access-register contents during a return operation.

Elsewhere, at column 16, lines 17 *et seq.*, Baum further describes:

There are three types of entries in the linkage stack: header entries 40 having a backward pointer, trailer entries 42 having a forward pointer, and state entries 43 (see linkage stack section 36). A header entry and a trailer entry are at the beginning and end, respectively, of a linkage-stack section, and are used to chain linkage-stack sections together.

Thus, Baum does describe “save the execution state,” “a backward pointer,” and “chain linkage-stack sections together.” However, Applicants find the description to be fundamentally different from the recited arrangement and believe that Baum lacks “checking, in backward order . . . for the value of a variable in the chain of state frames” as recited in claim 1.

**Further, Baum’s description of features related to a linkage stack does not anticipate the recited “a cached value copied from a prior frame in the chain of state frames.”** Even if the Examiner disagrees regarding the “chain of state frames,” Applicants have further amended the claim to include a feature relating to caching. Not only is such a feature missing in Baum, but one of skill in the art would have no reason to apply caching to Baum’s linkage stack as recited.

Accordingly, claim 1 and its dependent claims, 3-6 and 30-31, are allowable over Baum.

#### Claim 7

Similarly, claim 7 recites “walking backward through the chain of state frames from a last state frame toward the first state frame until an update to the value for the variable in the chain of state frames is found” and “copying an update from a prior state frame to a current state frame, thereby speeding up further accesses to the variable.”

Accordingly, in addition to its own merits, claim 7 and its dependent claims, 32-34, are allowable for at least the same reasons given for claim 1.

Claim 38

Claim 38 mimics the language of claim 7. Accordingly, in addition to its own merits, claim 38 and its dependent claims, 39-41, are allowable for at least the same reasons given for claim 7.

Claim 42

Claim 42 mimics the language of claim 7. Accordingly, in addition to its own merits, claim 42 is allowable for at least the same reasons given for claim 7.

Dependent Claims

In the interest of brevity, Applicants do not argue the language of the individual dependent claims, other than to point out that each recites a patentably distinct, novel, and non-obvious combination that stands on its own merits in addition to the reasons given for the independent claims.

***Interview Request***

If any issues remain, the Examiner is requested to contact the undersigned attorney by telephone for resolution.

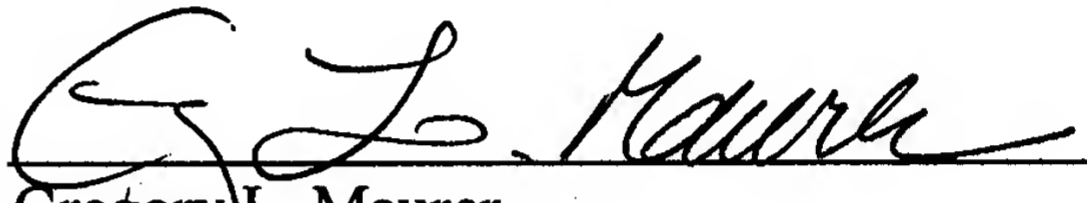
***Conclusion***

The claims in their present form stand ready for allowance. Such action is respectfully requested.

Respectfully submitted,

KLARQUIST SPARKMAN, LLP

One World Trade Center, Suite 1600  
121 S.W. Salmon Street  
Portland, Oregon 97204  
Telephone: (503) 595-5300  
Facsimile: (503) 595-5301

By   
Gregory L. Maurer  
Registration No. 43,781